# Edge Boxes: Locating Object Proposals from Edges
# 边框：从边缘定位对象建议

论文：https://pdollar.github.io/files/papers/ZitnickDollarECCV14edgeBoxes.pdf (https://pdollar.github.io/files/papers/ZitnickDollarECCV14edgeBoxes.pdf)




## Abstract
## 摘要

The use of object proposals is an effective recent approach for increasing the computational efficiency of object detection. We propose a novel method for generating object bounding box proposals using edges. Edges provide a sparse yet informative representation of an image. Our main observation is that the number of contours that are wholly contained in a bounding box is indicative of the likelihood of the box containing an object. We propose a simple box objectness score that measures the number of edges that exist in the box minus those that are members of contours that overlap the box's boundary. Using efficient data structures, millions of candidate boxes can be evaluated in a fraction of a second, returning a ranked set of a few thousand top-scoring proposals. Using standard metrics, we show results that are significantly more accurate than the current state-of-the-art while being faster to compute. In particular, given just 1000 proposals we achieve over 96% object recall at overlap threshold of 0.5 and over 75% recall at the more challenging overlap of 0.7. Our approach runs in 0.25 seconds and we additionally demonstrate a near real-time variant with only minor loss in accuracy.
对象提议的使用是提高对象检测的计算效率的有效的近期方法。我们提出了一种使用边生成对象边界框提议的新方法。边缘提供图像的稀疏但信息丰富的表示。我们的主要观察结果是，完全包含在边界框中的轮廓数量表示包含物体的盒子的可能性。我们提出了一个简单的盒子对象度分数，用于测量盒子中存在的边缘数量减去与盒子边界重叠的轮廓成员的边缘数量。使用高效的数据结构，可以在几分之一秒内评估数百万个候选框，返回几千个最高评分提案的排名集。使用标准指标，我们显示的结果比当前最先进的技术更加准确，同时计算速度更快。特别是，仅给出1000个提议，我们在重叠阈值为0.5时实现超过96％的对象回忆，在0.7的更具挑战性的重叠时回忆超过75％。我们的方法在0.25秒内运行，我们还演示了近实时变量，精度损失很小。


Keywords: object proposals, object detection, edge detection
关键词：对象提议，对象检测，边缘检测

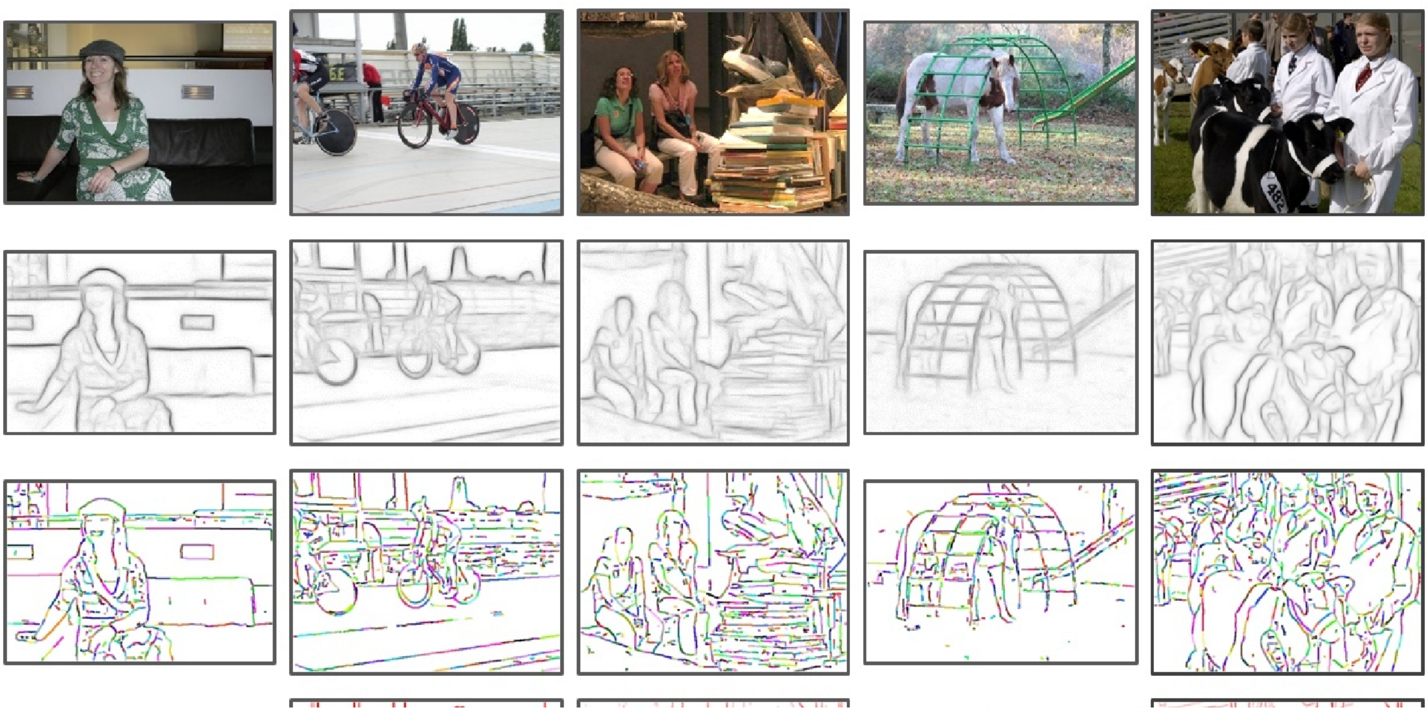
## 1 Introduction
## 1简介

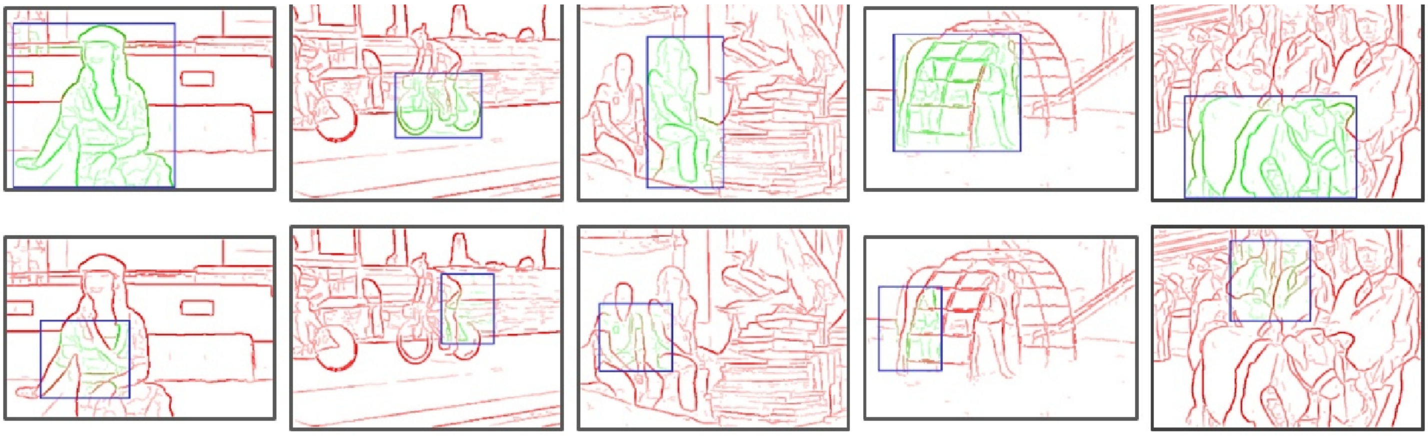
The goal of object detection is to determine whether an object exists in an image, and if so where in the image it occurs. The dominant approach to this problem over the past decade has been the sliding windows paradigm in which object classification is performed at every location and scale in an image [1– 3]. Recently, an alternative framework for object detection has been proposed. Instead of searching for an object at every image location and scale, a set of object bounding box proposals is first generated with the goal of reducing the set of positions that need to be further analyzed. The remarkable discovery made by these approaches [4–11] is that object proposals may be accurately generated in a manner that is agnostic to the type of object being detected. Object proposal generators are currently used by several state-of-the-art object detection algorithms [5, 12, 13], which include the winners of the 2013 ImageNet detection challenge [14] and top methods on the PASCAL VOC dataset [15].

对象检测的目标是确定对象是否存在于图像中，如果存在，则在图像中出现的位置。在过去十年中，这个问题的主要方法是滑动窗口范例，其中对象分类在图像中的每个位置和比例进行[1-3]。最近，已经提出了用于物体检测的替代框架。不是在每个图像位置和比例处搜索对象，而是首先生成一组对象边界框提议，其目的是减少需要进一步分析的位置集。通过这些方法[4-11]所做出的显着发现是，可以以与被检测对象的类型无关的方式准确地生成对象提议。对象提议生成器目前由几种最先进的对象检测算法[5,12,13]使用，其中包括2013 ImageNet检测挑战的获胜者[14]和PASCAL VOC数据集的顶级方法[15] 。

High recall and $efficiency$ are critical properties of an object proposal generator. If a proposal is not generated in the vicinity of an object that object Fig. 1. Illustrative examples showing from top to bottom (first row) original image, (second row) Structured Edges [16], (third row) edge groups, (fourth row) example correct bounding box and edge labeling, and (fifth row) example incorrect boxes and edge labeling. Green edges are predicted to be part of the object in the box ( $w_b$ ( $s_i$ ) = 1), while red edges are not ( $w_b$ ( $s_i$ ) = 0). Scoring a candidate box based solely on the number of contours it wholly encloses creates a surprisingly effective object proposal measure. The edges in rows 3-5 are thresholded and widened to increase visibility.

高召回率和$efficiency$是对象提议生成器的关键属性。如果在对象（图1）的对象附近没有生成提议。从上到下（第一行）显示原始图像的说明性示例，（第二行）结构化边[16]，（第三行）边缘组，（第四行）示例正确边界框和边缘标记，以及（第五行）示例不正确盒子和边缘标签。预计绿色边缘是框中对象的一部分（$w_b$（$s_i$）= 1），而红色边缘不是（$w_b$（$s_i$）= 0）。仅根据其完全包含的轮廓数量对候选框进行评分会产生令人惊讶的有效对象建议度量。行3-5中的边缘被阈值化并加宽以增加可视性。





cannot be detected. An effective generator is able to obtain high recall using a relatively modest number of candidate bounding boxes, typically ranging in the hundreds to low thousands per image. The precision of a proposal generator is less critical since the number of generated proposals is a small percentage of the total candidates typically considered by sliding window approaches (which may evaluate tens to hundreds of thousands of locations per object category). Since object proposal generators are primarily used to reduce the computational cost of the detector, they should be significantly faster than the detector itself. There is some speculation that the use of a small number of object proposals may even improve detection accuracy due to reduction of spurious false positives [4].
无法检测到。有效的生成器能够使用相对适中数量的候选边界框获得高召回率，通常范围在每个图像数百到数千。提案生成器的精度不太重要，因为生成的提议的数量是通常通过滑动窗口方法考虑的候选总数的一小部分（其可以评估每个对象类别数十到数十万个位置）。由于对象建议生成器主要用于降低探测器的计算成本，因此它们应该比探测器本身快得多。有人猜测，由于虚假误报的减少，使用少量对象提案甚至可能提高检测准确度[4]。

In this paper we propose Edge Boxes, a novel approach to generating object bounding box proposals directly from edges. Similar to segments, edges provide a simplified but informative representation of an image. In fact, line drawings of an image can accurately convey the high-level information contained in an image using only a small fraction of the information [17, 18]. As we demonstrate, the use of edges offers many computational advantages since they may be efficiently computed [16] and the resulting edge maps are sparse. In this work we investigate how to directly detect object proposals from edge-maps.
在本文中，我们提出了Edge Boxes，一种直接从边缘生成对象边界框提议的新方法。与片段类似，边缘提供简化但信息丰富的图像表示。实际上，图像的线条图可以仅使用一小部分信息准确地传达图像中包含的高级信息[17,18]。正如我们所展示的那样，边缘的使用具有许多计算优势，因为它们可以被有效地计算[16]并且得到的边缘图是稀疏的。在这项工作中，我们研究如何从边缘图直接检测对象提议。

Our main contribution is the following observation: the number of contours wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object. We say a contour is wholly enclosed by a box if all edge pixels belonging to the contour lie within the interior of the box. Edges tend to correspond to object boundaries, and as such boxes that tightly enclose a set of edges are likely to contain an object. However, some edges that lie within an object's bounding box may not be part of the contained object. Specifically, edge pixels that belong to contours straddling the box's boundaries are likely to correspond to objects or structures that lie outside the box, see Figure 1. In this paper we demonstrate that scoring a box based on the number of contours it wholly encloses creates a surprisingly effective proposal measure. In contrast, simply counting the number of edge pixels within the box is not as informative. Our

approach bears some resemblance to superpixels straddling measure introduced by [4]; however, rather than measuring the number of straddling contours we instead remove such contours from consideration.
我们的主要贡献是以下观察：由边界框完全包围的轮廓的数量表示包含物体的盒子的可能性。我们说如果属于轮廓的所有边缘像素都位于盒子的内部，则轮廓完全被盒子包围。边缘倾向于对应于对象边界，因此紧密围绕一组边缘的框可能包含对象。但是，位于对象边界框内的某些边可能不是所包含对象的一部分。具体而言，属于跨越框边界的轮廓的边缘像素可能对应于位于框外的对象或结构，请参见图1。在本文中，我们证明了根据它完全包含的轮廓数量对盒子进行评分，可以创建一个令人惊讶的有效提议措施。相反，简单地计算框内边缘像素的数量并不具有信息性。我们的方法与[4]引入的跨尺度跨度测量方法有一些相似之处;然而，我们不是考虑跨越轮廓的数量，而是从考虑中去除这些轮廓。

As the number of possible bounding boxes in an image is large, we must be able to score candidates efficiently. We utilize the fast and publicly available Structured Edge detector recently proposed in [16, 19] to obtain the initial edge map. To aid in later computations, neighboring edge pixels of similar orientation are clustered together to form groups. Affinities are computed between edge groups based on their relative positions and orientations such that groups forming long continuous contours have high affinity. The score for a box is computed by summing the edge strength of all edge groups within the box, minus the strength of edge groups that are part of a contour that straddles the box’s boundary, see Figure 1.
由于图像中可能的边界框的数量很大，我们必须能够有效地对候选人进行评分。我们利用最近在[16,19]中提出的快速且公开可用的结构化边缘检测器来获得初始边缘图。为了有助于以后的计算，将相似取向的相邻边缘像素聚集在一起以形成组。基于边缘组的相对位置和取向计算边缘组之间的效率，使得形成长连续轮廓的组具有高的效率。通过对框内所有边缘组的边缘强度求和，减去作为跨越框边界的轮廓的一部分的边缘组的强度来计算框的得分，参见图1。

We evaluate candidate boxes utilizing a sliding window approach, similar to traditional object detection. At every potential object position, scale and aspect ratio we generate a score indicating the likelihood of an object being present. Promising candidate boxes are further refined using a simple coarse-tofine search. Utilizing efficient data structures, our approach is capable of rapidly finding the top object proposals from among millions of potential candidates.
我们使用滑动窗口方法评估候选框，类似于传统的对象检测。在每个潜在的物体位置，比例和纵横比，我们生成一个分数，指示物体存在的可能性。使用简单的粗到细搜索进一步确定有希望的候选框。利用高效的数据结构，我们的方法能够从数百万潜在候选者中快速找到最佳对象提案。

We show improved recall rates over state-of-the-art methods for a wide range of intersection over union thresholds, while simultaneously improving efficiency. In particular, on the PASCAL VOC dataset [15], given just 1000 proposals we achieve over 96% object recall at overlap threshold of 0.5 and over 75% recall at an overlap of 0.7. At the latter and more challenging setting, previous stateof-the-art approaches required considerably more proposals to achieve similar recall. Our approach runs in quarter of a second, while a near real-time variant runs in a tenth of a second with only a minor loss in accuracy.
我们展示了针对各种交叉点超过联合阈值的最先进方法的召回率，同时提高了效率。特别是在PASCAL VOC数据集[15]中，仅给出了1000个提案，我们在重叠阈值为0.5时实现超过96％的对象回忆，在重叠为0.7时回忆超过75％。在后者和更具挑战性的环境中，先前最先进的方法需要更多的建议来实现类似的召回。我们的方法在四分之一秒内运行，而近乎实时的变体在十分之一秒内运行，精度仅有轻微损失。

# 2 Related work

# 2相关工作

The goal of generating object proposals is to create a relatively small set of candidate bounding boxes that cover the objects in the image. The most common use of the proposals is to allow for efficient object detection with complex and expensive classifiers [5, 12, 13]. Another popular use is for weakly supervised learning [20, 21], where by limiting the number of candidate regions, learning with less supervision becomes feasible. For detection, recall is critical and thousands of candidates can be used, for weakly supervised learning typically a few hundred proposals per image are kept. Since it's inception a few years ago [4, 9, 6], object proposal generation has found wide applicability.
生成对象提议的目的是创建一组覆盖图像中对象的相对较小的候选边界框。这些建议最常见的用途是允许使用复杂且昂贵的分类器进行高效的物体检测[5,12,13]。另一种流行的用途是弱监督学习[20,21]，其中通过限制候选区域的数量，用较少的监督进行学习变得可行。对于检测，召回是关键的并且可以使用数千个候选者，对于弱监督学习，通常每个图像保留几百个提议。自几年前[4,9,6]开始以来，对象提议生成已经广泛应用。

Generating object proposals aims to achieve many of the benefits of image segmentation without having to solve the harder problem of explicitly partitioning an image into non-overlapping regions. While segmentation has found limited success in object detection [22], in general it fails to provide accurate object regions. Hoiem et al. [23] proposed to use multiple overlapping segmentations to overcome errors of individual segmentations, this was explored further by [24] and [25] in the context of object detection. While use of multiple segmentations improves robustness, constructing coherent segmentations is an inherently difficult task. Object proposal generation seeks to sidestep the challenges of full segmentation by directly generating multiple overlapping object proposals.
生成对象提议旨在实现图像分割的许多好处，而不必解决将图像明确划分为非重叠区域的更难的问题。虽然分割在物体检测中发现了有限的成功[22]，但一般来说它无法提供准确的物体区域。Hoiem等人。 [23]建议使用多个重叠分割来克服个别分割的错误，[24]和[25]在物体检测的背景下进一步探讨了这一点。虽然使用多个分段可以提高稳健性，但构建相干分段本身就是一项困难的任务。对象提议生成旨在通过直接生成多个重叠对象提议来回避完全分割的挑战。

Three distinct paradigms have emerged for object proposal generation. Candidate bounding boxes representing object proposals can be found by measuring their 'objectness' [4, 11], producing multiple foreground-background segmentations of an image [6, 9, 10], or by merging superpixels [5, 8]. Our approach provides an alternate framework based on edges that is both simpler and more efficient while sharing many advantages with previous work. Below we briefly outline representative work for each paradigm; we refer readers to Hosang et al. [26] for a thorough survey and evaluation of object proposal methods.
针对对象提议生成出现了三种不同的范例。表示对象提议的候选边界框可以通过测量它们的"对象性"[4,11]，产生图像的多个前景 - 背景分割[6,9,10]，或者通过合并超像素[5,8]来找到。我们的方法提供了一个基于边缘的替代框架，既简单又高效，同时与以前的工作共享许多优点。下面我们简要介绍每个范例的代表性工作;我们将读者推荐给Hosang等人。 [26]对对象提案方法进行彻底调查和评估。

Objectness Scoring: Alexe et al. [4] proposed to rank candidates by combining a number of cues in a classification framework and assigning a resulting 'objectness' score to each proposal. [7] built on this idea by learning efficient cascades to more quickly and accurately rank candidates. Among multiple cues, both [4] and [7] define scores based on edge distributions near window boundaries. However, these edge scores do not remove edges belonging to contours

intersecting the box boundary, which we found to be critical. [4] utilizes a superpixel straddling measure penalizing candidates containing segments overlapping the boundary. In contrast, we suppress straddling contours by propagating information across edge groups that may not directly lie on the boundary. Finally, recently [11] proposed a very fast objectness score based on image gradients.
对象评分：Alexe等。 [4]建议通过在分类框架中组合多个线索并为每个提议分配得到的“对象性”得分来对候选人进行排名。 [7]建立在这个想法的基础上，通过学习有效的级联来更快，更准确地对候选人进行排名。在多个线索中，[4]和[7]都基于窗口边界附近的边缘分布来定义得分。但是，这些边缘分数不会删除属于与框边界相交的轮廓的边，我们发现这些边是关键的。 [4]利用超像素跨骑测量惩罚包含与边界重叠的段的候选者。相反，我们通过跨越可能不直接位于边界上的边缘组传播信息来抑制跨越轮廓。最后，最近[11]提出了一种基于图像梯度的非常快的物体得分。

Seed Segmentation: [6, 9, 10] all start with multiple seed regions and generate a separate foreground-background segmentation for each seed. The primary advantage of these approaches is generation of high quality segmentation masks, the disadvantage is their high computation cost (minutes per image).
种子分割：[6,9,10]都以多个种子区域开始，并为每个种子生成单独的前景 - 背景分割。这些方法的主要优点是生成高质量的分割掩模，缺点是它们的计算成本高（每个图像的分钟数）。

Superpixel Merging: Selective Search [5] is based on computing multiple hierarchical segmentations based on superpixels from [27] and placing bounding boxes around them. Selective Search has been widely used by recent top detection methods [5, 12, 13] and the key to its success is relatively fast speed (seconds per image) and high recall. In a similar vein, [8] propose a randomized greedy algorithm for computing sets of superpixels that are likely to occur together. In our work, we operate on groups of edges as opposed to superpixels. Edges can be represented probabilistically, have associated orientation information, and can be linked allowing for propagation of information; properly exploited, this additional information can be used to achieve large gains in accuracy.
超像素合并：选择性搜索[5]基于计算基于[27]的超像素的多层次分割并在它们周围放置边界框。选择性搜索已被最近的顶级检测方法[5,12,13]广泛使用，其成功的关键是相对较快的速度（每个图像的秒数）和高召回率。类似地，[8]提出了一种随机贪婪算法，用于计算可能一起出现的超像素集。在我们的工作中，我们操作边缘组而不是超像素。边缘可以概率地表示，具有相关的取向信息，并且可以链接以允许信息的传播;如果利用得当，这些附加信息可用于实现准确性的大幅提升。

As far as we know, our approach is the first to generate object bounding box proposals directly from edges. Unlike all previous approaches we do not use segmentations or superpixels, nor do we require learning a scoring function from multiple cues. Instead we propose to score candidate boxes based on the number of contours wholly enclosed by a bounding box. Surprisingly, this conceptually simple approach out-competes previous methods by a significant margin.
据我们所知，我们的方法是首先直接从边缘生成对象边界框提议。与以前的所有方法不同，我们不使用分段或超像素，也不需要从多个线索中学习评分函数。相反，我们建议根据边界框完全包围的轮廓数来对候选框进行评分。令人惊讶的是，这种概念上简单的方法与以前的方法相比具有显着的优势。

## 3 Approach
## 3方法

In this section we describe our approach to finding object proposals. Object proposals are ranked based on a single score

computed from the contours wholly enclosed in a candidate bounding box. We begin by describing a data structure based on edge groups that allows for efficient separation of contours that are fully enclosed by the box from those that are not. Next, we define our edge-based scoring function. Finally, we detail our approach for finding top-ranked object proposals that uses a sliding window framework evaluated across position, scale and aspect ratio, followed by refinement using a simple coarse-to-fine search.
在本节中，我们将介绍我们发现对象提议的方法。对象提议基于从完全包含在候选边界框中的轮廓计算的单个分数来排名。我们首先描述一个基于边缘组的数据结构，该数据结构允许完全包围盒子的轮廓与非盒子完全包围的轮廓。接下来，我们定义基于边缘的评分函数。最后，我们详细介绍了我们的方法，用于查找排名靠前的对象提议，这些提议使用在位置，比例和纵横比上评估的滑动窗口框架，然后使用简单的粗到细搜索进行重新设置。

Given an image, we initially compute an edge response for each pixel. The edge responses are found using the Structured Edge detector [16, 19] that has shown good performance in predicting object boundaries, while simultaneously being very efficient. We utilize the single-scale variant with the sharpening enhancement introduced in [19] to reduce runtime. Given the dense edge responses, we perform Non-Maximal Suppression (NMS) orthogonal to the edge response to find edge peaks, Figure 1. The result is a sparse edge map, with each pixel p having an edge magnitude $m_p$ and orientation $\theta_p$. We define edges as pixels with $m_p > 0.1$ (we threshold the edges for computational efficiency). A contour is defined as a set of edges forming a coherent boundary, curve or line.
给定图像，我们最初计算每个像素的边缘响应。使用结构化边缘检测器[16,19]找到边缘响应，该检测器在预测对象边界方面表现出良好的性能，同时非常高效。我们利用[19]中引入的锐化增强的单尺度变体来减少运行时间。给定密集边缘响应，我们执行与边缘响应正交的非最大抑制（NMS），以找到边缘峰值，如图1所示。结果是稀疏边缘图，每个像素p具有边缘幅度$m_p$和方向$\theta_p$。我们将边缘定义为具有$m_p > 0.1$的像素（我们将边缘阈值用于计算效率）。轮廓被定义为形成相干边界，曲线或线的一组边。

3.1 Edge groups and affinities
3.1边缘组和一个功能

As illustrated in Figure 1, our goal is to identify contours that overlap the bounding box boundary and are therefore unlikely to belong to an object contained by the bounding box. Given a box b, we identify these edges by computing for each $p \in b$ with $m_p > 0.1$ its maximum affinity with an edge on the box boundary. Intuitively, edges connected by straight contours should have high affinity, where those not connected or connected by a contour with high curvature should have lower affinity. For computational efficiency we found it advantageous to group edges that have high affinity and only compute affinities between edge groups. We form the edge groups using a simple greedy approach that combines 8-connected edges until the sum of their orientation differences is above a threshold ( $\pi / 2$). Small groups are merged with neighboring groups. An illustration of the edge groups is shown in Figure 1, row 3.
如图1所示，我们的目标是识别与边界框边界重叠的轮廓，因此不太可能属于边界框所包含的对象。给定一个方框b，我们通过计算每个$p \in b$来识别这些边缘，其中$m_p >$为其最大边界，边框位于边界上。直观地，由直线轮廓连接的边缘应该具有高的刚度，其中那些未通过具有高曲率的轮廓连接或连接的边缘应该具有较低的效率。为了计算效率，我们发现对具有高效率的边缘进行分组并且仅计算边缘组之间的效率是有利的。我们使用简单的贪婪方法形成边缘组，该方法结合了8个连接边，直到它们的方向差的总和高于阈值（$\pi / 2$）。小团体与邻近团体合并。边缘组的图示如图1第3行所示。

Given a set of edge groups $s_i \in S$ , we compute an affinity between each pair of neighboring groups. For a pair of

groups $s_i$ and $s_j$, the affinity is computed based on their mean positions $x_i$ and $x_j$ and mean orientations $\theta_i$ and $\theta_j$. Intuitively, edge groups have high affinity if the angle between the groups' means in similar to the groups' orientations. Specifically, we compute the affinity a( $s_i, s_j$ ) using:
给定一组边缘组$s_i \in S$，我们计算每对相邻组之间的效率。对于一对组$s_i$和$s_j$，根据它们的平均位置$x_i$和$x_j$以及平均方向$\theta_i$和$\theta_j$来计算效率。直观地，如果组之间的角度意味着与组的方向相似，则边缘组具有高的效率。具体来说，我们使用以下方法计算一个函数a（$s_i, s_j$）：

$$a(s_i, s_j) = |\cos(\theta_i - \theta_{ij}) \cos(\theta_j - \theta_{ij})|^{\gamma}, \tag{1}$$

where $\theta_{ij}$ is the angle between $x_i$ and $x_j$. The value of γ may be used to adjust the affinity's sensitivity to changes in orientation, with γ = 2 used in practice. If two edge groups are separated by more than 2 pixels their affinity is set to zero. For increased computational efficiency only affinities above a small threshold (0.05) are stored and the rest are assumed to be zero.
其中$\theta_{ij}$是$x_i$和$x_j$之间的角度。γ的值可用于调整效率对方向变化的敏感度，实际使用γ= 2。如果两个边缘组间隔超过2个像素，则其效果将设置为零。为了提高计算效率，仅存储高于小阈值（0.05）的函数，并假设其余的为零。

The edge grouping and affinity measure are computationally trivial. In practice results are robust to the details of the edge grouping.
边缘分组和效率测量在计算上是微不足道的。实际上，结果对于边缘分组的细节是稳健的。

3.2 Bounding box scoring
3.2边界框评分

Given the set of edge groups S and their affinities, we can compute an object proposal score for any candidate bounding box b. To find our score, we first compute the sum $m_i$ of the magnitudes $m_p$ for all edges p in the group $s_i$. We also pick an arbitrary pixel position $\bar{x}_i$ of some pixel p in each group $s_i$. As we will show, the exact choice of $p \in s_i$ does not matter.
给定一组边缘组S及其一个函数，我们可以计算任何候选边界框b的对象建议分数。为了找到我们的分数，我们首先计算$s_i$组中所有边p的$m_p$的总和$m_i$。我们还在每个组$s_i$中选择一些像素p的任意像素位置$\bar{x}_i$。正如我们将要展示的那样，$p \in s_i$的确切选择并不重要。

For each group $s_i$ we compute a continuous value $w_b$ ( $s_i$ ) ∈ [0, 1] that indicates whether $s_i$ is wholly contained in b, $w_b$ ( $s_i$ ) = 1, or not, $w_b$ ( $s_i$ ) = 0. Let $S_b$ be the set of edge groups that overlap the box b's boundary. We find $S_b$ using an efficient data structure that is described in Section 3.3. For all $s_i \in S_b$, $w_b$ ( $s_i$ ) is set to 0. Similarly $w_b$ ( $s_i$ ) = 0 for all $s_i$ for which $\bar{x}_i \notin b$, since all of its pixels must either be outside of b or $s_i \in S_b$. For the remaining edge groups for which $\bar{x}_i \in b$ and $s_i \notin S_b$ we compute $w_b$ ( $s_i$ ) as follows:
对于每个组$s_i$，我们计算连续值$w_b$（$s_i$）∈[0,1]，指示$s_i$是否完全包含在b，$w_b$（$s_i$）= 1中，$w_b$（$s_i$）= 0。设$S_b$是与框b边界重叠的边集组。我们使用第3.3节中描述的高效数据结构找到$S_b$。对于所有$s_i \in S_b$，$w_b$（$s_i$）设置为0。类似地$w_b$（$s_i$）= 0表示所有$s_i$的$\bar{x}_i \notin b$，因为它的所有像素都必须在b或$s_i \in S_b$之外。对于其中为$\bar{x}_i \in b$和$s_i \notin S_b$的剩余边缘组，我们计算$w_b$（$s_i$），如下所示：

$|T|-1$

$$w_b(s_i) = 1 - \max_T \prod_j^{|T|-1} a(t_j, t_{j+1}), \tag{2}$$

where T is an ordered path of edge groups with a length of $|T|$ that begins with some t1 $\in S_b$ and ends at $t_{|T|}$ = $s_i$. If no such path exists we define $w_b$ ( $s_i$ ) = 1. Thus, Equation (2) finds the path with highest affinity between the edge group $s_i$ and an edge group that overlaps the box's boundary. Since most pairwise affinities are zero, this can be done efficiently.
其中T是边长组的有序路径，其长度为$|T|$，以某些t1 $\in S_b$开头，结束于$t_{|T|}$ = $s_i$。如果不存在这样的路径，我们定义$w_b$（$s_i$）= 1。因此，等式（2）找到边缘组$s_i$与重叠框边界的边缘组之间具有最高效率的路径。由于大多数成对的效率为零，因此可以有效地完成。

Using the computed values of $w_b$ we define our score using:
使用$w_b$的计算值，我们使用以下方法定义我们的分数：

$$h_b = \frac{\sum_i w_b(s_i) m_i}{2(b_w + b_h)^\kappa}, \tag{3}$$

where $b_w$ and $b_w$ are the box's width and height. Note that we divide by the box's perimeter and not its area, since edges have a width of one pixel regardless of scale. Nevertheless, a value of κ = 1.5 is used to offset the bias of larger windows having more edges on average.
其中$b_w$和$b_w$是框的宽度和高度。请注意，我们除以框的周长而不是其面积，因为无论比例如何，边都具有一个像素的宽度。然而，使用κ= 1.5的值来设置平均具有更多边缘的较大窗口的偏差。

In practice we use an integral image to speed computation of the numerator in Equation (3). The integral image is used to compute the sum of all $m_i$ for which ¯ $\bar{x}_i \in b$. Next, for all $s_i$ with ¯ $\bar{x}_i \in b$ and $w_b$ ( $s_i$ ) < 1, (1 $-$ $w_b$ ( $s_i$ )) $m_i$ is subtracted from this sum. This speeds up computation considerably as typically $w_b$ ( $s_i$ ) = 1 for most $s_i$ and all such $s_i$ do not need to be explicitly considered. Finally, it has been observed that the edges in the center of the box are of less importance than those near the box's edges [4]. To account for this observation we can subtract the edge magnitudes from a box $b^{in}$ centered in b:
实际上，我们使用积分图像来加速公式（3）中分子的计算。积分图像用于计算$\bar{x}_i \in b$的所有$m_i$的总和。接下来，对于具有¯$\bar{x}_i \in b$和$w_b$（$s_i$）<1的所有$s_i$，（1 $-$ $w_b$（$s_i$））从该总和中减去$m_i$。这大大加快了计算速度，因为大多数$s_i$通常$w_b$（$s_i$）= 1，并且不需要明确考虑所有这样的$s_i$。最后，已经观察到盒子中心的边缘不如盒子边缘附近的边缘重要[4]。为了解释这个观察，我们可以从以b为中心的$b^{in}$框中减去边缘幅度：

$$h_b^{in} = h_b - \frac{\sum_{p \in b^{in}} m_p}{2(b_w + b_h)^\kappa}, \tag{4}$$

where the width and height of $b^{in}$ is $b_w/$ 2 and $b_h/$ 2 respectively. The sum of the edge magnitudes in $b^{in}$ can be efficiently computed using an integral image. As shown in Section 4 we found $h_b^{in}$ offers slightly better accuracy than $h_b$ with minimal additional computational cost.
其中$b^{in}$的宽度和高度分别为$b_w/$ 2和$b_h/$ 2。$b^{in}$中边缘幅度的总和可以使用积分图像进行有效计算。如第4节所

示，我们发现$h_b^{in}$的精度略高于$h_b$，并且计算成本最低。

### 3.3 Finding intersecting edge groups

### 3.3查找相交的边缘组

In the previous section we assumed the set of edge groups $S_b$ that overlap the box b's boundary is known. Since we evaluate a huge number of bounding boxes (Sec. 3.4), an efficient method for finding $S_b$ is critical. Naive approaches such as exhaustively searching all of the pixels on the boundary of a box would be prohibitively expensive, especially for large boxes.

在上一节中，我们假设边框组$S_b$与框b的边界重叠是已知的。由于我们评估了大量的边界框（第3.4节），因此找到$S_b$的一种有效方法至关重要。诸如彻底搜索盒子边界上的所有像素的朴素方法将是非常昂贵的，特别是对于大盒子而言。

We propose an efficient method for finding intersecting edge groups for each side of a bounding box that relies on two additional data structures. Below, we describe the process for finding intersections along a horizontal boundary from pixel (c0 $, r$ ) to (c1 $, r$ ). The vertical boundaries may be handled in a similar manner. For horizontal boundaries we create two data structures for each row of the image. The first data structure stores an ordered list $L_r$ of edge group indices for row r. The list is created by storing the order in which the edge groups occur along the row r. An index is only added to $L_r$ if the edge group index changes from one pixel to the next. The result is the size of $L_r$ is much smaller than the width of the image. If there are pixels between the edge groups that are not edges, a zero is added to the list. A second data structure $K_r$ with the same size as the width of the image is created that stores the corresponding index into $L_r$ for each column c in row r. Thus, if pixel p at location ( $c, r$ ) is a member of edge group $s_i$ , $L_r$ ( $K_r$ (c)) = i. Since most pixels do not belong to an edge group, using these two data structures we can efficiently find the list of overlapping edge groups by searching $L_r$ from index $K_r$ (c0) to $K_r$ (c1).

我们提出了一种有效的方法，用于为边界框的每一边找到相交的边缘组，该边界组依赖于两个额外的数据结构。下面，我们描述了从像素（c0 $, r$）到（c1 $, r$）沿水平边界找到交点的过程。可以以类似的方式处理垂直边界。对于水平边界，我们为图像的每一行创建两个数据结构。第一个数据结构存储行r的边缘组索引的有序列表$L_r$。通过存储沿着行r出现边缘组的顺序来创建列表。如果边缘组索引从一个像素更改为下一个像素，则仅将索引添加到$L_r$。结果是$L_r$的尺寸远小于图像的宽度。如果边缘组之间的像素不是边缘，则将零添加到列表中。创建具有与图像宽度相同大小的第二数据结构$K_r$，其将对应的索引存储到行r中的每个列c的$L_r$中。因此，如果位置处的像素p（$c, r$）是边缘组$s_i$的成员，则$L_r$（$K_r$（c））= i。由于大多数像素不属于边缘组，因此使用这两种数据结构，我们可以通过从索引$K_r$（c0）到$K_r$（c1）搜索$L_r$，有效地找到重叠边缘组的列表。

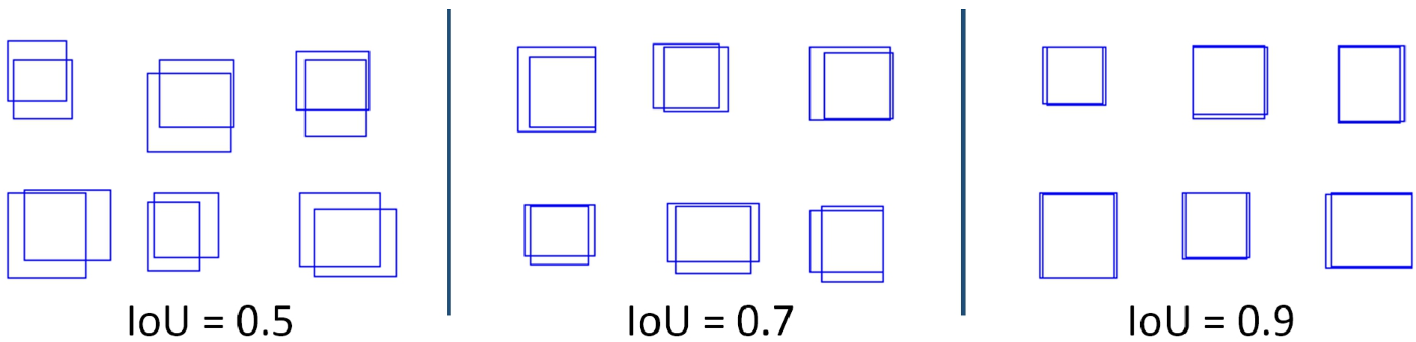



Fig. 2. An illustration of random bounding boxes with Intersection over Union (IoU) of 0.5, 0.7, and 0.9. An IoU of 0.7

provides a reasonable compromise between very loose (IoU of 0.5) and very strict (IoU of 0.9) overlap values.
图2.具有0.5,0.7和0.9的联合交叉（IoU）的随机边界框的图示。IoU为0.7提供了非常宽松（IoU为0.5）和非常严格（IoU为0.9）重叠值之间的合理折衷。

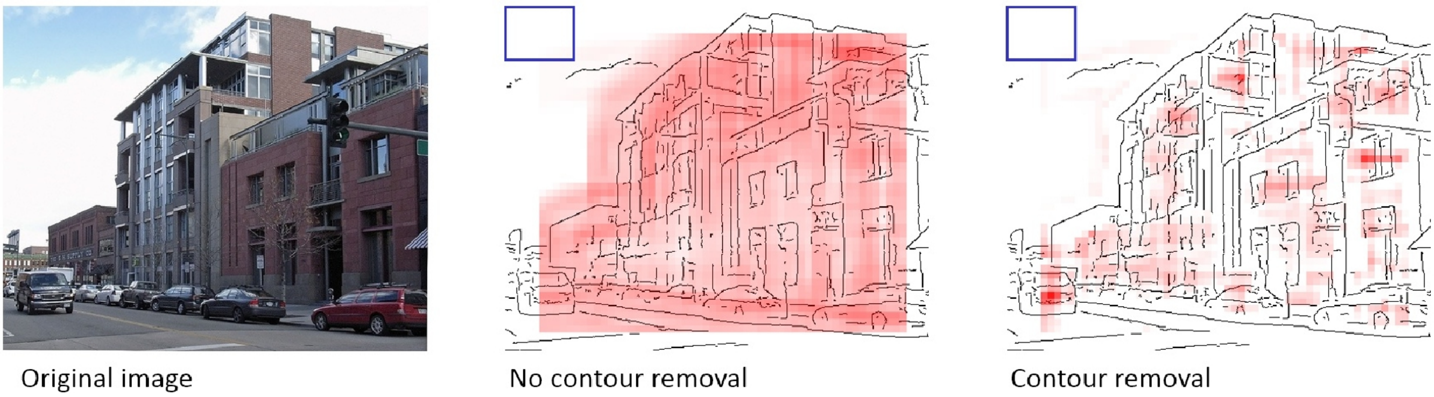
3.4 Search strategy
3.4搜索策略

When searching for object proposals, the object detection algorithm should be taken into consideration. Some detection algorithms may require object proposals with high accuracy, while others are more tolerant of errors in bounding box placement. The accuracy of a bounding box is typically measured using the Intersection over Union (IoU) metric. IoU computes the intersection of a candidate box and the ground truth box divided by the area of their union. When evaluating object detection algorithms, an IoU threshold of 0.5 is typically used to determine whether a detection was correct [15]. However as shown in Figure 2, an IoU score of 0.5 is quite loose. Even if an object proposal is generated with an IoU of 0.5 with the ground truth, the detection algorithm may provide a low score. As a result, IoU scores of greater than 0.5 are generally desired.
在搜索对象提议时，应考虑对象检测算法。一些检测算法可能需要具有高准确度的对象提议，而其他检测算法则更容忍边界框放置中的错误。通常使用Union Union（IoU）度量标准来测量边界框的准确性。IoU计算候选框和地面实况框的交集，除以它们的并集区域。在评估对象检测算法时，通常使用0.5的IoU阈值来确定检测是否正确[15]。但是，如图2所示，IoU得分为0.5非常宽松。即使使用具有基础事实的IoU为0.5生成对象提议，检测算法也可以提供低分。结果，通常需要大于0.5的IoU分数。

In this section we describe an object proposal search strategy based on the desired IoU, δ, for the detector. For high values of δ we generate a more concentrated set of bounding boxes with higher density near areas that are likely to contain an object. For lower values of δ the boxes can have higher diversity, since it is assumed the object detector can account for moderate errors in box location. Thus, we provide a tradeoff between finding a smaller number of objects with higher accuracy and a higher number of objects with less accuracy. Note that previous methods have an implicit bias for which δ they are designed for, e.g. Objectness [4] and Randomized Prim [8] are tuned for low and high δ, respectively, whereas we provide explicit control over diversity versus accuracy. We begin our search for candidate bounding boxes using a sliding window search over position, scale and aspect ratio. The step size for each is determined using a single parameter α indicating the IoU for neighboring boxes. That is, the step sizes in translation, scale and aspect ratio are determined such that one step results in neighboring boxes having an IoU of α. The scale values range from a minimum box area of σ = 1000 pixels to the full image. The aspect ratio varies from $1/\tau$ to τ , where τ = 3 is used in practice. As we discuss in Section 4, a value of α = 0.65 is ideal for most values of δ. However, if a highly accurate $\delta >$ 0.9 is required, α may be increased to 0.85.
在本节中，我们描述了基于探测器所需IoU，δ的对象建议搜索策略。对于δ的高值，我们在靠近可能包含物体的区域附近产生更集中的边界框，其具有更高的密度。对于较低的δ值，盒子可以具有更高的多样性，因为假设物体检测器可以解决盒子位置中的中等误差。因此，我们提供了一个交换ff，在较少数量的具有较高准确度的对象和较少数量的对象之间进行精确度较低。注意，先前的方法具有隐式偏差，其设计为δ，例如，对象性[4]和随机化Prim [8]分别针对低和高δ进行了调整，而我们提供了对多样性与准确性的明确控制。我们使用滑动窗口搜索位置，比例和纵横比开始搜索候选边界框。使用指示相邻框的IoU的单个参数α来确定每个的步长。也就是说，确定平移，比例和纵横比的步长，使得一步导致具有α的IoU的相邻框。比例值的范围从σ= 1000像素的最小框区域到

完整图像。纵横比从1 $/\tau$变化到τ，其中τ= 3在实践中使用。正如我们在第4节中讨论的那样，对于大多数δ值，α= 0.65的值是理想的。但是，如果需要高精度的$\delta > 0.9$，则α可能会增加到0.85。

After a sliding window search is performed, all bounding box locations with a score $h_b^{in}$ above a small threshold are refined. Refinement is performed using Fig. 3. Illustration of the computed score using (middle) and removing (right) contours that overlap the bounding box boundary. Notice the lack of clear peaks when the contours are not removed. The magnitudes of the scores are normalized for viewing. The box dimensions used for generating the heatmaps are shown by the blue rectangles.
在执行滑动窗口搜索之后，将重新定义具有高于小阈值的分数$h_b^{in}$的所有边界框位置。使用图3执行改进。使用（中间）和移除（右）轮廓与边界框边界重叠的计算得分的图示。请注意，未去除轮廓时缺少明显的峰。分数的大小被标准化以供观看。用于生成热图的框尺寸由蓝色矩形表示。




a greedy iterative search to maximize $h_b^{in}$ over position, scale and aspect ratio. After each iteration, the search step is reduced in half. The search is halted once the translational step size is less than 2 pixels.
贪婪的迭代搜索，以最大化$h_b^{in}$的位置，比例和宽高比。每次迭代后，搜索步骤减少一半。一旦平移步长小于2个像素，搜索就会停止。

Once the candidate bounding boxes are refined, their maximum scores are recorded and sorted. Our final stage performs Non-Maximal Suppression (NMS) of the sorted boxes. A box is removed if its IoU is more than β for a box with greater score. We have found that in practice setting β = δ + 0.05 achieves high accuracy across all values of δ, Section 4.
一旦重新定义了候选边界框，就会记录和排序它们的最大分数。我们的最终阶段执行排序框的非最大抑制（NMS）。如果对于具有更高分数的框，其IoU大于β，则移除框。我们已经发现，在实践中，设定β=δ+ 0.05在δ，第4节的所有值上实现了高精度。

# 4 Results
# 4结果

In this section we explore the performance and accuracy of our Edge Boxes algorithm in comparison to other approaches. Following the experimental setup of previous approaches [7, 5, 8, 4] we evaluate our algorithm on the PASCAL VOC 2007 dataset [15]. The dataset contains 9,963 images. All results on variants of our approach are reported on the validation set and our results compared to other approaches are reported on the test set.
在本节中，我们将探讨Edge Boxes算法与其他方法相比的性能和准确性。在先前方法[7,5,8,4]的实验设置之后，我们在PASCAL VOC 2007数据集上评估我们的算法[15]。该数据集包含9,963个图像。我们的方法变体的所有结

果都在验证集上报告，我们在测试集上报告了与其他方法相比的结果。

## 4.1 Approach variants

## 4.1方法变体

We begin by testing various variants of our approach on the validation set. Figure 4(a, b) illustrates the algorithm's behavior based on the parameters α and β that control the step size of the sliding window search and the NMS threshold, respectively, when generating 1000 object proposals.

我们首先在验证集上测试我们方法的各种变体。图4（a，b）示出了当生成1000个对象提议时，基于分别控制滑动窗口搜索的步长和NMS阈值的参数α和β的算法行为。

As α is increased, the density of the sampling is increased, resulting in more candidate boxes being evaluated and slower runtimes, Table 1. Notice that the results for α = 0.65 are better than or nearly identical to α = 0.70 and α = 0.75. Thus, if a lower IoU value δ is desired α = 0.65 provides a nice accuracy vs. efficiency tradeoff. Depending on the desired IoU value of δ, the value of β may be adjusted accordingly. A value of β = δ + 0.05 achieves high accuracy across all desired δ. As shown in Table 1, changes in β have minimal effect on runtime.

随着α增加，采样密度增加，导致评估更多的候选框和更慢的运行时间，表1。注意，α= 0.65的结果优于或几乎相同于α= 0.70和α= 0.75。因此，如果期望较低的IoU值δ，则α= 0.65提供了与效率交易ff相比较好的准确度。根据所需的δ的IoU值，可以相应地调整β的值。β=δ+ 0.05的值在所有期望的δ上实现高精度。如表1所示，β的变化在运行时具有最小的影响。

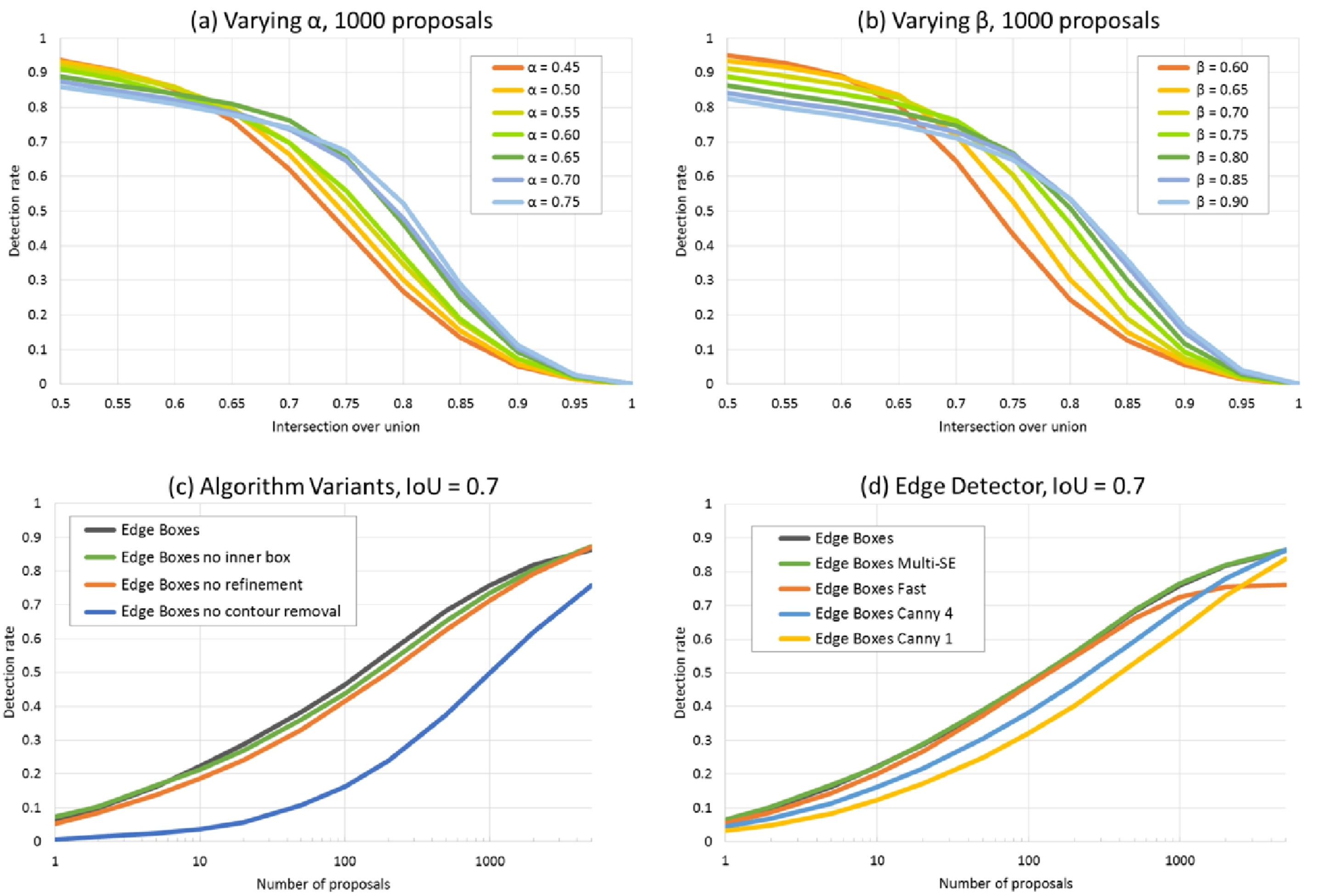


Fig. 4. A comparison of various variants of our approach. (a) The detection rate when varying the parameter α that varies the density of the sampling rate (default α = 0.65). (b) Results while varying the parameter β controlling the NMS threshold (default β = 0.75). (c) The detection accuracy when various stages are removed from the algorithm, including the removal of edges in the inner box, the bounding box location refinement, and the removal of the contours that overlap the box's boundaries. (d) Detection accuracy when different edge detectors are used, including single-scale Structured Edges [16] (default), multi-scale Structure Edges, and a fast variant that runs at 10 fps without the edge sharpening enhancement introduced in [19]. Results using the Canny edge detector [28] with varying amounts of blur are also shown.
图4.我们方法的各种变体的比较。（a）改变改变采样率密度的参数α时的检测率（默认值α= 0.65）。（b）改变控制NMS阈值的参数β的结果（默认β= 0.75）。（c）从算法中移除各个阶段时的检测精度，包括去除内盒中的边缘，边界框位置的改进，以及移除与盒子边界重叠的轮廓。（d）使用不同边缘检测器时的检测精度，包括单刻度结构边[16]（默认），多尺度结构边缘，以及在[19]中引入的没有边缘锐化增强的情况下以10 fps运行的快速变体。还显示了使用具有不同模糊量的Canny边缘检测器[28]的结果。

Three useful variants of our algorithm are shown in Table 1; Edge Boxes 50, Edge Boxes 70, and Edge Boxes 90 that have settings for α and β adjusted for IoU thresholds of δ = 0.5, 0.7 and 0.9 respectively. For higher IoU thresholds that require extremely tight bounding boxes, α must be adjusted to search more densely resulting in longer runtimes. Otherwise, α may be kept fixed.
我们算法的三个有用变体如表1所示;边框50，边框70和边框90具有α和β的设置，分别针对δ= 0.5,0.7和0.9的IoU阈值进行调整。对于需要极其紧密的边界框的更高IoU阈值，必须调整α以更密集地搜索，从而导致更长的运行时间。否则，α可以保持固定。

Our second set of experiments tests several variants of the algorithm, Figure 4(c, d). For these experiments we set δ to an intermediate value of 0.7 and show detection rates when varying the number of object proposals. The primary contribution of our paper is that contours that overlap the bounding box's boundary should be removed when computing the box's score. Figure 4 shows that if these contours are not removed a significant drop in accuracy is observed. To gain intuition into the effect of the contour removal on the score, we illustrate the score computed with and without contour removal in Figure 3. With contour removal the scores have strong peaks around clusters of edges that are more likely to form objects given the current box's size. Notice the strong peak with contour removal in the bottom left-hand corner corresponding to the van. When contours are not removed strong responses are observed everywhere.
我们的第二组实验测试了算法的几种变体，图4（c，d）。对于这些实验，我们将δ设置为0.7的中间值，并在改变对象提议的数量时显示检测率。我们论文的主要贡献是，在计算盒子的分数时，应该删除与边界框边界重叠的轮廓。图4显示，如果不去除这些轮廓，则观察到精度显着下降。为了获得对得分上轮廓去除效果的直觉，我们在图3中说明了有和没有轮廓去除的计算得分。通过轮廓去除，得分在边缘簇周围具有强峰，在给定当前盒子的大小的情况下更可能形成对象。注意在与货车相对应的左下角的轮廓消除的强峰。当没有去除轮廓时，到处都会观察到强烈的反应。

| | IoU = 0.5 | | IoU = 0.7 | | IoU = 0.9 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | AUC | Recall | AUC | Recall | AUC | Recall | Runtime | α | β |
| Edge boxes 50 | **.64** | **96%** | .36 | 55% | .04 | 5% | **.25s** | .65 | .55 |
| Edge boxes 70 | .58 | 89% | **.45** | **76%** | .06 | 9% | **.25s** | .65 | .75 |
| Edge boxes 90 | .38 | 59% | .28 | 46% | **.15** | **28%** | 2.5s | .85 | .95 |

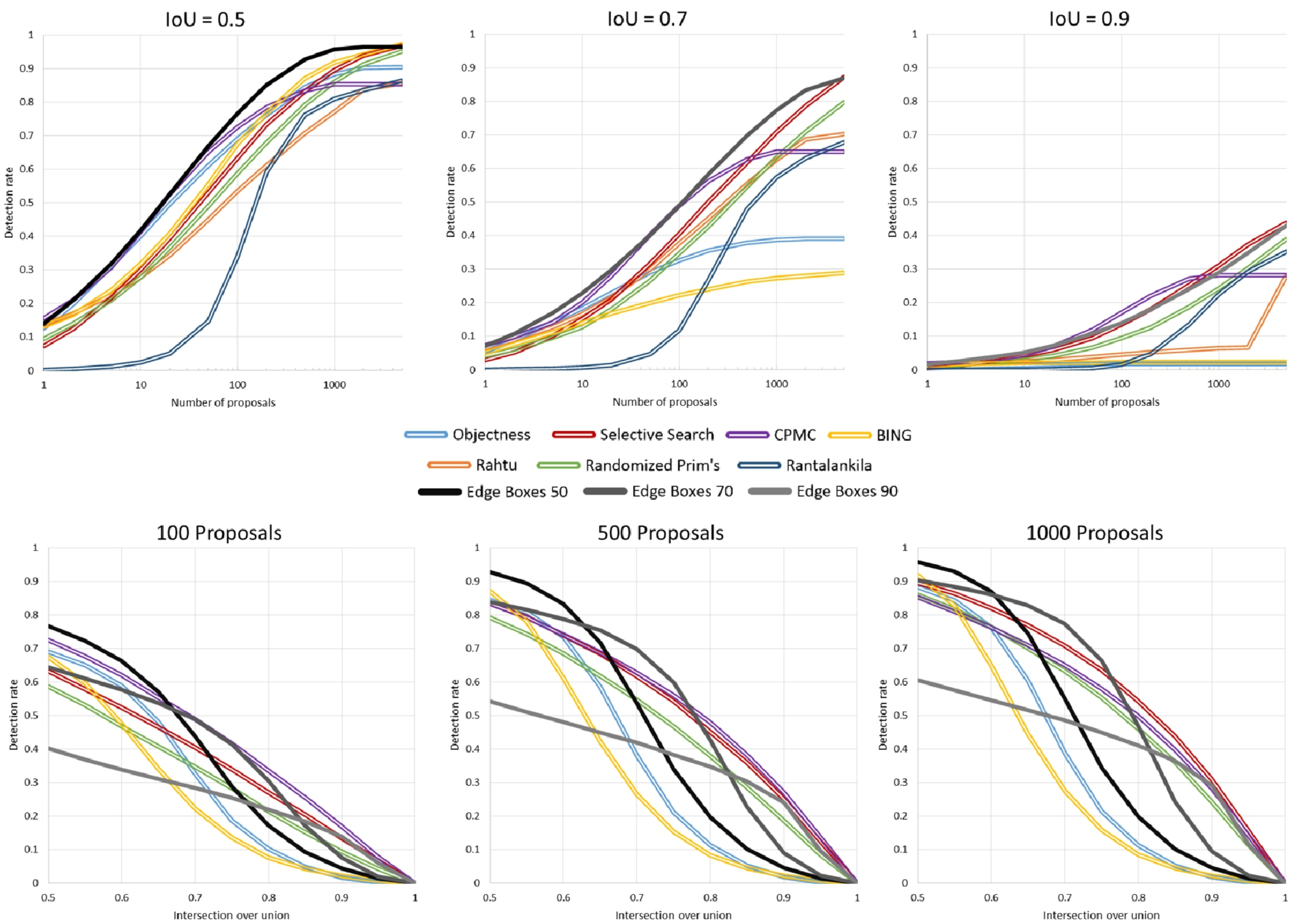
*Table 1. Accuracy measures and runtimes for three variants of our algorithm: Edge Boxes 50, Edge Boxes 70 and Edge Boxes 90. Accuracy measures include Area Under the Curve (AUC) and proposal recall at 1000 proposals. Parameter values for α and β are shown. All other parameters are held constant.*
*表1.我们算法的三种变体的精度测量和运行时间：Edge Boxes 50，Edge Boxes 70和Edge Boxes 90。准确度测量包括曲线下面积（AUC）和1000个提案的提案召回。显示了α和β的参数值。所有其他参数保持不变。*

If the center edges are not removed, using $h_b$ instead of $h_b^{in}$ , a small drop in accuracy is found. Not performing box location refinement results in a more significant drop in accuracy. The quality of the initial edge detector is also important, Figure 4(d). If the initial edge map is generated using gradient-based Canny edges [28] with varying blur instead of Structured Edges [16] the results degrade. If multi-scale Structured Edges are computed instead of single-scale edges there is a minimal gain in accuracy. Since single-scale edges can be computed more efficiently we use single-scale edges for all remaining experiments. The runtime of our baseline approach which utilizes single-scale edges is 0.25s. If near real-time performance is desired, the parameters of the algorithm may be adjusted to return up to 1000 boxes with only a minor loss in accuracy. Specifically, we can reduce α to 0.625, and increase the threshold used to determine which boxes to refine from 0.01 to 0.02. Finally, if we also disable the sharpening enhancement of the Structured Edge detector [19], the runtime of our algorithm is 0.09s. As shown in Figure 4(d), this variant, called Edge Boxes Fast, has nearly identical results when returning fewer than 1000 boxes.
如果未移除中心边缘，则使用$h_b$而不是$h_b^{in}$，会发现精度略有下降。不执行盒位置修改会导致准确性下降更多。初始边缘检测器的质量也很重要，如图4（d）所示。如果使用基于梯度的Canny边[28]生成初始边缘图，其中模糊变化而不是结构边[16]，则结果会降低。如果计算多尺度结构化边缘而不是单尺度边缘，则精度增益最小。由于可以更有效地计算单尺度边缘，因此我们对所有剩余实验使用单尺度边缘。使用单尺度边缘的基线方法的运行时间为0.25秒。如果需要接近实时性能，则可以调整算法的参数以返回多达1000个盒子，仅具有微小的精度损失。具体而言，我们可以将α减小到0.625，并增加用于确定从0.01到0.02重新确定哪些框的阈值。最后，如果我们也禁用结构化边缘检测器的锐化增强[19]，我们算法的运行时间为0.09s。如图4（d）所示，这种称为Edge Boxes Fast的变体在返回少于1000个盒子时具有几乎相同的结果。

4.2 Comparison with state-of-the-art
4.2与最先进的技术进行比较

We compare our Edge Boxes algorithm against numerous state-of-the-art algorithms summarized in Table 2. Results of all competing methods were provided by Hosang et al. [26] in a standardized format. Figure 5 (top) shows the detection rates when varying the number of object proposals for different IoU thresholds. For each plot, we update our parameters based on the desired value of δ using the parameters in Table 1. Edge Boxes performs well across all IoU values and for both a small and large number of candidates. Selective Search [5] achieves competitive accuracy, especially at higher IoU values and larger number of boxes. CPMC [6] generates high quality proposals but produces relatively few candidates and is thus unable to achieve high recall. BING [11], which is very fast, generates only very loosely fitting proposals and hence is only competi Fig. 5. Comparison of Edge Boxes to various state-of-the-algorithms, including Objectness [4], Selective Search [5], Randomized Prim's [8] and Rahtu [7]. The variations of our algorithm are tested using δ = 0.5, 0.7 and 0.9 indicated by Edge Boxes 50, Edge Boxes 70 and Edge Boxes 90. (top) The detection rate vs. the number of bounding box proposals for various intersection over union thresholds. (bottom) The detection rate vs. intersection over union for various numbers of object proposals.

我们将Edge Boxes算法与表2中总结的众多最先进算法进行了比较。所有竞争方法的结果由Hosang等人提供。[26]采用标准格式。图5（上）显示了在针对不同的IoU阈值改变对象提议的数量时的检测率。对于每个图，我们使用表1中的参数基于所需的δ值更新我们的参数。Edge Boxes在所有IoU值以及少量和大量候选者中表现良好。选择性搜索[5]实现了竞争准确性，尤其是在更高的IoU值和更大数量的盒子时。CPMC [6]提出了高质量的提案，但产生的候选人数相对较少，因此无法实现高召回率。BING [11]非常快，只产生非常松散的建议，因此只有竞争对手图5。边缘框与各种算法状态的比较，包括Objectness [4]，Selective Search [5]，Randomized Prim [8]和Rahtu [7]。我们的算法的变化是使用由边框50，边框70和边框90指示的δ= 0.5,0.7和0.9来测试的。（上）检测率与各种交叉超过联合阈值的边界框建议的数量。（下）不同数量的对象提案的检测率与联合的交集。




tive at low IoU. In contrast our approach achieves good results across a variety of IoU thresholds and quantity of object proposals. In fact, as shown in Table 2, to achieve a recall of 75% with an IoU of 0.7 requires 800 proposals using Edge Boxes, 1400 proposals using Selective Search, and 3000 using Randomized Prim's. No other methods achieve 75% recall using even 5000 proposals. Edge Boxes also achieves a significantly higher maximum recall (87%) and Area Under the Curve (AUC = 0.46) as compared to all approaches except Selective Search.

低IoU时。相比之下，我们的方法在各种IoU阈值和对象提案数量方面取得了良好的效果。实际上，如表2所示，要实现75％的IoU召回率，需要使用Edge Box的800个提案，使用选择性搜索的1400个提案和使用Randomized Prim的3000个提案。没有其他方法甚至可以使用5000个提案实现75％的召回。与除选择性搜索之外的所有方法相比，Edge Boxes还实现了显着更高的最大召回率（87％）和曲线下面积（AUC = 0.46）。

Figure 5 (bottom) shows the detection rate when varying the IoU threshold for different numbers of proposals. Similar to Figure 4(a,b), these plots demonstrate that setting parameters based on δ, the desired IoU threshold, leads to good performance. No single algorithm or set of parameters is capable of achieving superior performance across all IoU thresholds. However, Edge Boxes 70 performs well over a wide range of IoU thresholds that are typically desired in practice (IoU between 0.5 and 0.8, Figure 2). Segmentation based methods along with Edge Boxes 90 perform best at very high IoU values.
图5（下图）显示了针对不同数量的提案改变IoU阈值时的检测率。与图4（a，b）类似，这些图表明，基于δ（所需的IoU阈值）设置参数可以获得良好的性能。没有一种算法或一组参数能够在所有IoU阈值上实现卓越的性能。然而，Edge Boxes 70在很宽的IoU阈值范围内表现良好，这在实践中通常是期望的（IoU在0.5和0.8之间，图2）。基于分段的方法以及Edge Boxes 90在非常高的IoU值下表现最佳。

We compare the runtime and summary statistics of our approach to other methods in Table 2. The runtimes for Edge Boxes includes the 0.1 seconds needed
我们将我们的方法的运行时和汇总统计数据与表2中的其他方法进行比较。Edge Boxes的运行时间包括0.1秒

AUC N@25% N@50% N@75% Recall Time
AUC N @ 25％N @ 50％N @ 75％召回时间

| BING [11] | .20 | 292 | – | – | 29% | **.2s** |
|---|---|---|---|---|---|---|

Rantalankila [10] .23 184 584 – 68% 10s
海滩银行[10] .23 184 584 - 68％10s

Objectness [4] .27 27 – – 39% 3s
对象性[4] .27 27 - - 39％3s

Rand. Prim’s [8] .35 42 349 3023 80% 1s
兰德。 Prim的[8] .35 42 349 3023 80％1s

Rahtu [7] .37 29 307 – 70% 3s
Rahtu [7] .37 29 307 - 70％3s

Selective Search [5] .40 28 199 1434 87% 10s
选择性搜索[5] .40 28 199 1434 87％10s

CPMC [6] .41 15 111 – 65% 250s
CPMC [6] .41 15 111 - 65％250s

| Edge boxes 70 | **.46** | **12** | **108** | **800** | **87%** | **.25s** |
|---|---|---|---|---|---|---|

Table 2. Results for our approach, Edge Boxes 70, compared to other methods for IoU threshold of 0.7. Methods are sorted by increasing Area Under the Curve (AUC). Additional metrics include the number of proposals needed to achieve 25%, 50% and 75% recall and the maximum recall using 5000 boxes. Edge Boxes is best or near best under every metric. All method runtimes were obtained from [26].
表2.与IoU阈值为0.7的其他方法相比，我们的方法Edge Boxes 70的结果。通过增加曲线下面积（AUC）对方法进

行分类。其他指标包括实现25％，50％和75％召回所需的提案数量以及使用5000个盒子进行的最大召回。边框在每个指标下都是最佳或接近最佳的。所有方法运行时间均来自[26]。

to compute the initial edges. Table 2 shows that our approach is significantly faster and more accurate than previous approaches. The only methods with comparable accuracy are Selective Search and CPMC, but these are considerably slower. The only method with comparable speed is BING, but BING has the worst accuracy of all evaluated methods at IoU of 0.7.
计算初始边缘。表2显示我们的方法比以前的方法显着更快，更准确。唯一具有可比精度的方法是选择性搜索和CPMC，但这些方法要慢得多。唯一具有可比速度的方法是BING，但BING在IoU为0.7时所有评估方法的准确度最差。

Finally, qualitative results are shown in Figure 6. Many of the errors occur with small or heavily occluded objects in the background of the images.
最后，定性结果如图6所示。许多错误发生在图像背景中的小的或严重遮挡的对象中。

## 5 Discussion
## 5讨论

In this paper we propose an effective method for finding object proposals in images that relies on one simple observation: the number of edges that are wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object. We describe a straightforward scoring function that computes the weighted sum of the edge strengths within a box minus those that are part of a contour that straddles the box's boundary. Using efficient data structures and smart search strategies we can find object proposals rapidly. Results show both improved accuracy and increased efficiency over the state of the art.
在本文中，我们提出了一种有效的方法，用于在图像中找到依赖于一个简单观察的对象提议：由边界框完全包围的边数表示包含对象的框的可能性。我们描述了一个简单的评分函数，该函数计算框内边缘强度的加权和减去横跨框边界的轮廓的一部分。使用高效的数据结构和智能搜索策略，我们可以快速找到对象提案。结果表明，与现有技术相比，提高了准确度并提高了效率。

One interesting direction for future work is using the edges to help generate segmentation proposals in addition to the bounding box proposals for objects. Many edges are removed when scoring a candidate bounding box; the location of these suppressed edges could provide useful information in generating segmentations. Finally we will work with Hosang at al. to add Edge Boxes to their recent survey and evaluation of object proposal methods [26] and we also hope to evaluate our proposals coupled with state-of-the-art object detectors [13].
未来工作的一个有趣方向是使用边缘来帮助生成分段提议以及对象的边界框提议。在对候选边界框进行评分时，会删除许多边缘;这些被抑制的边缘的位置可以在生成分割时提供有用的信息。最后，我们将与Hosang合作。将边缘框添加到他们最近的对象提议方法的调查和评估中[26]，我们也希望评估我们的建议以及最先进的物体探测器[13]。

Source code for Edge Boxes will be made available online.
Edge Boxes的源代码将在线提供。

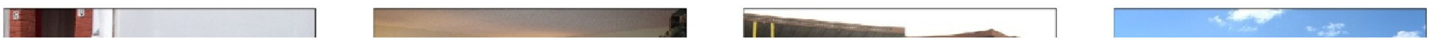


Fig. 6. Qualitative examples of our object proposals. Blue bounding boxes are the closest produced object proposals to each ground truth bounding box. Ground truth bounding boxes are shown in green and red, with green indicating an object was found and red indicating the object was not found. An IoU threshold of 0.7 was used to determine correctness for all examples. Results are shown for Edge Boxes 70 with 1,000 object proposals. At this setting our approach returns over 75% of object locations.
图6.我们的对象提案的定性示例。蓝色边界框是最接近每个地面真实边界框的生成对象建议。地面实况边界框以绿色和红色显示，绿色表示找到对象，红色表示未找到对象。使用IoU阈值0.7来确定所有示例的正确性。边框70显示了具有1,000个对象提议的结果。在此设置下，我们的方法返回超过75％的对象位置。

# References
# 参考

1. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV 57(2) (2004) 137–154
1. Viola，P.A.，Jones，M.J。：强大的实时人脸检测。 IJCV 57（2）（2004）137-154

2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)

2. Dalal，N.，Triggs，B。：用于人体检测的定向梯度的直方图。在：CVPR。（2005年）

3. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI 32(9) (2010) 1627–1645
3. Felzenszwalb，P.，Girshick，R.，McAllester，D.，Ramanan，D。：使用有区别训练的部分模型进行物体检测。 PAMI 32（9）（2010）1627-1645

4. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. PAMI 34(11) (2012)
4. Alexe，B.，Deselaers，T.，Ferrari，V。：测量图像窗口的对象性。 PAMI 34（11）（2012）

5. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. IJCV (2013)
5. Uijlings，J.R.R。，van de Sande，K.E.A。，Gevers，T.，Smeulders，A.W.M。：选择性搜索物体识别。 IJCV（2013）

6. Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. PAMI 34(7) (2012)
6. Carreira，J.，Sminchisescu，C。：Cpmc：使用约束参数最小割的自动对象分割。 PAMI 34（7）（2012）

7. Rahtu, E., Kannala, J., Blaschko, M.: Learning a category independent object detection cascade. In: ICCV. (2011)
7. Rahtu，E.，Kannala，J.，Blaschko，M。：学习类别独立的物体检测级联。在：ICCV。（2011年）

8. Manen, S., Guillaumin, M., Van Gool, L., Leuven, K.: Prime object proposals with randomized prims algorithm. In: ICCV. (2013)
8. Manen，S.，Guillaumin，M.，Van Gool，L.，Leuven，K。：使用随机prims算法的Prime对象提议。在：ICCV。（2013）

9. Endres, I., Hoiem, D.: Category-independent object proposals with diverse ranking. PAMI (2014)
9. Endres，I.，Hoiem，D。：具有不同排名的类别独立对象提案。 PAMI（2014）

10. Rantalankila, P., Kannala, J., Rahtu, E.: Generating object segmentation proposals using global and local search. In: CVPR. (2014)
10. Rantalankila，P.，Kannala，J.，Rahtu，E。：使用全局和局部搜索生成对象分割提议。在：CVPR。（2014）

11. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: BING: Binarized normed gradients for objectness estimation at 300fps. In: CVPR. (2014)
11. Cheng，M.M.，Zhang，Z.，Lin，W.Y.，Torr，P。：BING：用于300fps的物体估计的二值化标准梯度。在：CVPR。（2014）

12. Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: ICCV. (2013)
12. Wang，X.，Yang，M.，Zhu，S.，Lin，Y。：用于通用对象检测的Regionlets。在：ICCV。（2013）

13. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
13. Girshick，R.B.，Donahue，J.，Darrell，T.，Malik，J。：用于精确对象检测和语义分割的丰富特征层次结构。在：CVPR。（2014）

14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009)

15. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2) (2010) 303–338

15. Everingham，M.，Van Gool，L.，Williams，C.K.I.，Winn，J.，Zisserman，A。：pascal visual object classes（voc）挑战。 IJCV 88（2）（2010）303-338

16. Doll´ar, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV. (2013)

16. Doll'ar，P.，Zitnick，C.L。：用于快速边缘检测的结构化森林。在：ICCV。 （2013）

17. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information. Inc., New York, NY (1982)

17. Marr，D。：Vision：对人类表现和视觉信息处理的计算研究。 Inc.，纽约，纽约（1982）

18. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Transactions Graphics 31(4) (2012)

18. Eitz，M.，Hays，J.，Alexa，M。：人类如何描绘物体？ ACM Transactions图形31（4）（2012）

19. Doll´ar, P., Zitnick, C.L.: Fast edge detection using structured forests. CoRR **abs/1406.5549** (2014)

19. Doll'ar，P.，Zitnick，C.L。：使用结构化森林进行快速边缘检测。 CoRR **abs/1406.5549**（2014）

20. Deselaers, T., Alexe, B., Ferrari, V.: Localizing objects while learning their appearance. In: ECCV. (2010)

20. Deselaers，T.，Alexe，B.，Ferrari，V。：在学习外观的同时定位物体。在：ECCV。 （2010年）

21. Siva, P., Xiang, T.: Weakly supervised object detector learning with model drift detection. In: ICCV. (2011)

21. Siva，P.，Xiang，T。：用模型漂移检测进行弱监督物体探测器学习。在：ICCV。 （2011年）

22. Gu, C., Lim, J.J., Arbel´aez, P., Malik, J.: Recognition using regions. In: CVPR. (2009)

22. Gu，C.，Lim，J.J.，Arbel'aez，P.，Malik，J。：使用区域识别。在：CVPR。 （2009年）

23. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV. (2005)

23. Hoiem，D.，Efros，A.A.，Hebert，M。：单个图像的几何上下文。在：ICCV。 （2005年）

24. Russell, B.C., Freeman, W.T., Efros, A.A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR. (2006)

24. Russell，B.C.，Freeman，W.T.，Efros，A.A.，Sivic，J.，Zisserman，A。：使用多个分割来发现对象及其在图像集合中的范围。在：CVPR。 （2006年）

25. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: BMVC. (2007)

25. Malisiewicz，T.，Efros，A.A。：通过多个分割来改善对象的空间支持。在：BMVC。 （2007年）

26. Hosang, J., Benenson, R., Schiele, B.: How good are detection proposals, really? In: BMVC. (2014)

26. Hosang，J.，Benenson，R.，Schiele，B。：检测建议有多好，真的吗？在：BMVC。 （2014）

28. Canny, J.: A computational approach to edge detection. PAMI (6) (1986) 679–698

28. Canny，J。：边缘检测的计算方法。 PAMI（6）（1986）679-698